유전자 알고리즘 하드웨어 구현을 위한 전용 원칩 컴퓨터의 설계

박세현† · 이언학††

요 약

유전자 알고리즘(GA: Genetic Algorithm)은 다양한 영역에서 NP 문제를 해결하는 방법으로 알려져 있다. GA는 긴 연산 시간을 필요하다는 결점 때문에 최근 GA를 하드웨어로 구현하려는 연구가 주목 받아왔다. 본 논문은 GA의 하드웨어 구현을 위한 전용 원칩 컴퓨터를 제안한다. 제안된 전용 원칩 컴퓨터는16 비트 CPU core와 하드웨어 GA로 구성되어 있다.

기존의 하드웨어 GA는 GA의 처리하는데 있어서 메인 컴퓨터에 의존적이었으나 제안된 전용 원칩 컴퓨터는 메인 컴퓨터에 독립적이다. 또한 기존의 하드웨어 GA는 염색체의 길이가 고정되어 있는 데 비해 제안된 전용 원칩 컴퓨터의 염색체의 길이는 가변이며 16 비트 단위로 Pipeline 처리를 한다. 실험 결과는 제안된 원칩 컴퓨터가 랜덤 비트 동기 회로를 위한 진화 하드웨어 설계에 적용할 수 있다는 것을 보여준다.

Embedded One Chip Computer Design for Hardware Implementation of Genetic Algorithm

SeHyun Park† and UnHak Lee††

ABSTRACT

Genetic Algorithm(GA) has known as a method of solving NP problem in various applications. Since a major drawback of the GA is that it needs a long computation time, the hardware implementation of Genetic Algorithm is focused on in recent studies. This paper proposes a new type of embedded one chip computer for Hardware Implementation of Genetic Algorithm. The proposed embedded one chip computer consists of 16 Bit CPU core and hardware of genetic algorithm. In contrast to conventional hardware-oriented GA which is dependent on main computer in the process of GA, the proposed embedded one chip computer is independent on main computer. Conventional hardware GA uses the fixed length of chromosome but the proposed embedded one chip computer uses the variable length of chromosome by employing the efficient 16 bit Pipeline Unit.

Experimental results show that the proposed one chip computer is applicable to the design of evolvable hardware for Random NRZ bit synchronization circuit.

1. 서 론

유전자 알고리즘(GA:Genetic Algorithm)은 생물의 진화 원리로부터 착상된 확률적 탐색을 위한 기법이며 1975년에 소개된 홀랜드(Holland)의 저서 『Adaption in Natural and Artificial System』에 처음으로 소개되었다[1]. 1985년 유전자 알고리즘에 관한 국제회의(ICGA)가 개최된 이후 1990년부터 유전자 알고리즘에 대한 관심이 높아지기 시작하여 최근에는 패턴인식, 신경망 학습, 음성인식, 로봇제어 등의 분야에서 사용되고 있다. 유전자 알고리즘은 연산의 반복 수행으로 인한 속도의 저하가 심각한 문제가 되어왔으며 그에 대한 해결방안으로 병렬 분산 처리, 하드웨어 구현 등 여러 방법이 시도되었다.

† 정회원, 국립안동대학교 전자정보산업학부 부교수
†† 준회원, 안동대학교 대학원 전자공학과

유전자 알고리즘의 하드웨어의 구현은 Scott 등에 의해 제안된 간단한 유전자 알고리즘(Standard simple genetic algorithm)에 기초한 Hardware based Genetic Algorithm(이하 HGA)[2]가 그 시초이며 Yoshida 등은 이를 발전시켜 GAP(GA Processor)라는 하드웨어 지향의 GA를 제안하였다[3]. 또한 Wakabayashi 등은 범용 하드웨어로서 GAA (GA Accelerator) Chip이라 불리는 GA의 VLSI를 구현하였으나[4] 최근에는 Reprogramming이 가능한 FPGA를 사용한 구현이 국내·외 학계에서 연구되고 있다[5].

기존의 HGA는 주로 유전자 알고리즘의 수행 속도 개선을 위해 유전자 알고리즘을 FPGA등의 하드웨어로 구현 하든가 상용 프로세서를 이용하여 펌 웨어로 구현한다. 이러한 HGA는 GA를 위한 각종 파라미터 등을 설정하거나 염색체의 적합도를 구하기 위해서 메인 컴퓨터에 의존하게 되어있다. HGA가 메인 컴퓨터를 의존하게 되면 특정 분야에 대한 GA의 활용도가 떨어지게 된다. 특히 Evolvable Hardware (이하 EHW: Evolvable Hardware) 분야에 GA를 적용하기 위해서는 HGA가 메인 컴퓨터에 독립적이어야 할 필요가 있다.

또한 기존 HGA의 문제점은 32비트 혹은 64비트의 고정된 길이의 염색체에 대한 GA처리를 하였다. 그러나 이러한 고정된 길이의 염색체 처리는 실제의 GA 응용분야에 적용하는 데에는 부족한 크기일 뿐 아니라 고정된 길이의 염색체 처리는 효용가치가 없어 가변 길이의 염색체 처리를 필요로 한다.

따라서 본 논문에서는 GA의 처리를 위한 HGA의 하드웨어와 HGA의 제어를 위한 전용 CPU(중앙처리장치)를 인터페이스한 원칩 컴퓨터를 제안한다. 제안된 GA 처리를 위한 전용 원칩 컴퓨터는 기존의 하드웨어 HGA와 달리 가변 길이의 염색체를 16비트 단위로 Pipeline처리를 할 뿐 아니라 메인 컴퓨터에 독립적으로 동작하게 한다. 그리고 제안된 전용 원칩 컴퓨터를 EHW에 적용하고 그 효용성을 알아본다.

2. 유전자 이론

유전자 알고리즘은 기본적으로 Generate-and-Test형의 알고리즘으로서, Darwin의 적자 생존 이론을 기초로 한다. 즉, 문제에 대한 가능한 해들을 정해진 형태의 자료구조로 표현한 후 선택(selection), 교배(crossover), 돌연변이(mutation)의 세 가지 종류의 유전자 조작으로 이루어진 재생산(Reproduction) 과정을 반복 수행하여 개체내의 정보를 교환함으로서 최적의 해를 생산하는 알고리즘이다.

일반적으로 개체집단 $P(t) = \{x^t_1, \ldots, x^t_n\}$에서 유전자 알고리즘의 처리순서는 아래와 같다.

```
genetic algorithm
begin
  t ← 0
  initialize P(t)
  evaluate P(t)
  while (not termination-condition) do
    begin
      t ← t + 1
      select P(t) from P(t-1)
      crossover P(t)
      mutation P(t)
    end
end
```

a. 이산세대(discrete generation) 모델과 연속세대(continuous generation) 모델

유전자 알고리즘의 진화형태는 크게 이산세대 모델과 연속세대 모델 두 가지로 구분된다. 일반적으로 간단한 유전자 알고리즘에는 모든 개체가 일제히 자손을 만들고, 다음 세대의 집합을 만드는 이산세대 모델을 사용하고 있으며 또는 그것과는 반대로 각 세대에서 몇 개의 개체만 교체되는 연속세대 모델(continuous generation model)에 기반을 둔 것도 있다. 연속세대 모델은 De Jong[6]이 처음으로 제시한 것으로 현 세대 중에서 몇 퍼센트가 교체될 것인가를 나타내는 매개변수인 세대차이(generation gap)가 사용된다[7-10].

b. 선택(selection)

선택 연산자는 적합도가 높은 개체들은 살아남고 그렇지 못한 개체들은 도태되도록 유도함으로써 자연선택(natural selection)현상을 모델링 한다. 선택의 기반이 되는 것은 적합도 함수이며 여기에는 여러 가지 선택방법들이 존재하지만 그 기본 원리는 더 좋은 개체들에게 특권을 부여한다는 것에 있어서 공통적이다. 일반적으로 널리 사용되는 선택 방법으로

는 기본 모델인 룰렛 선택법(roulette selection), 적합도 비례 선택(fitness proportionate selection)과 기대치 선택법(expected-value selection), 순위 선택법(ranking selection), 토너먼트 선택법(tournament selection), 엘리트 보존 방식(elitist preserving selection)등이 있다[7-10,15].

c. 교배(crossover)

교배(crossover)는 두 부모의 염색체를 조합하고 바꾸어 자식의 염색체를 만드는 유전자 조작이다[7,9,10].

이러한 유전자 조작은 단순교배, 복수점 교배, 균일교배(uniform crossover)등이 있다.

d. 돌연변이(mutation)

돌연변이는 유전자를 일정한 확률로 변화시키는 유전자 조작이다. 돌연변이를 너무 큰 변이확률로 설정하면 스키마타가 전부 파괴되어 임의 탐색으로 변해버리게 되지만 국소 탐색에서 벗어나기 위해서 어느 정도의 변이는 필요하다. 즉, 돌연변이가 없는 경우에는 초기 유전자의 조합 이외의 공간을 탐색 할 수 없으며, 결국 찾고자하는 해의 질에도 한계가 드러난다. 일반적으로 돌연변이는 고정된 확률로 각 유전자가 변화하도록 설정하지만, 변이율을 동적으로 변화시키는 기법도 있다. 이러한 기법의 한가지로 적응변이(adaptive mutation)가 있다. 적응변이에는 교배의 결과, 만들어진 두 개 개체의 근사도를 해밍거리(hamming distance)로 측정하고, 거리가 가까울수록 높은 변이율로 하는 기법이다. 이것은 집단 중에서 유전자형의 다양성을 확보하고, 가능한 넓은 해공간을 탐색하고자 하는 의도이다[7-9].

3. 유전자 알고리즘을 처리를 위한 하드웨어와 전용 CPU의 설계

본 논문에서는 유전자 알고리즘처리를 위한 HGA를 설계하고 HGA을 제어할 전용 CPU를 인터페이스한 GA 처리를 위한 전용 원칩 컴퓨터를 설계한다. 설계된 전용 CPU는 외부 메모리로부터 염색체의 LOAD와 STORE를 위해 DMA를 사용하여 내부에는 교배부, 돌연변이부, 선택부를 하드웨어로 구성되어 있다. 그리고 염색체 보관은 외부 메모리에 저장하며 각각을 설명하면 다음과 같다.

3.1 전용 CPU

제안된 전용 CPU는 그림 1과 같은 기본 구조를 가지고 있으며 Altera사의 FLEX10K100ARC로 구현하였다. 전용 CPU는 내부 16비트의 버스를 가지고 있으며 내부에 프로그램 메모리와 데이터 메모리를 가진다. 염색체 저장을 위한 메모리는 외부에 두고 있고 DMA에 의해 구동시킨다. 실제의 GA 응용분야는 염색체의 길이가 길다. 특히 EHW 분야에 GA를 적용하려면 염색체의 표현에 있어서 대단히 많은 비트가 필요하다. 따라서 대규모 크기의 염색체를 병렬처리 할 수 있는 교배부와 돌연변이 부의 하드웨어를 설계한다는 것은 엄청난 하드웨어 자원이 요구되므로 비현실적이다.

따라서 본 논문에서는 염색체를 외부 대용량 메모리에 두고 DMA를 사용하녀 16비트 단위로 CPU에 이동시켜 Pipeline 처리를 한다. 제안된 전용 CPU Core의 목적은 GA의 처리 속도의 향상은 물론 HGA를 실제 GA 응용분야 특히 EHW 분야에 적용할 수 있도록 메인 컴퓨터에 독립적으로 동작할 수 있게 하는 것이다.

설계된 CPU는 소프트웨어 인터럽트와 하드웨어 인터럽트의 구조를 가지고 있으며 각각 4개의 인터럽트 소스를 받아들일 수 있다. 인터럽트 벡터는 각 염색체의 16비트 단위의 DMA 이동에 있어 인터럽트 처리를 이용할 수 있다. 그림 2는 인터럽트의 하드웨어 구조를 보여주고 있다.

3.2 교배부(crossover part)

그림 3은 구현된 GA의 교배부의 블록도 이다. Scott의 HGA, Yoshida의 GAP 그리고 Wakabayashi의 GA Chip에서는 일점 교배만을 사용하고 있으나 본 논문에서는 그림 3에 보는 바와 같이 Decoder 값에 따라 일점교배, 두점교배, 균일교배 등 필요에 따라 선택할 수 있도록 설계하였다. 그리고 이러한 선택은 교배 율과 함께 전용 CPU에서 설정할 수 있도록 설계하였다. 또한 염색체를 16비트 단위로 1 Clock에서 교배가 이루어지는 구조로 설계되었다.

3.3 돌연변이부(mutation part)

그림 4은 본 논문에서 설계한 돌연변이 부에 대한 블록도이다. 돌연변이부도 교배부와 같은 기능의 비
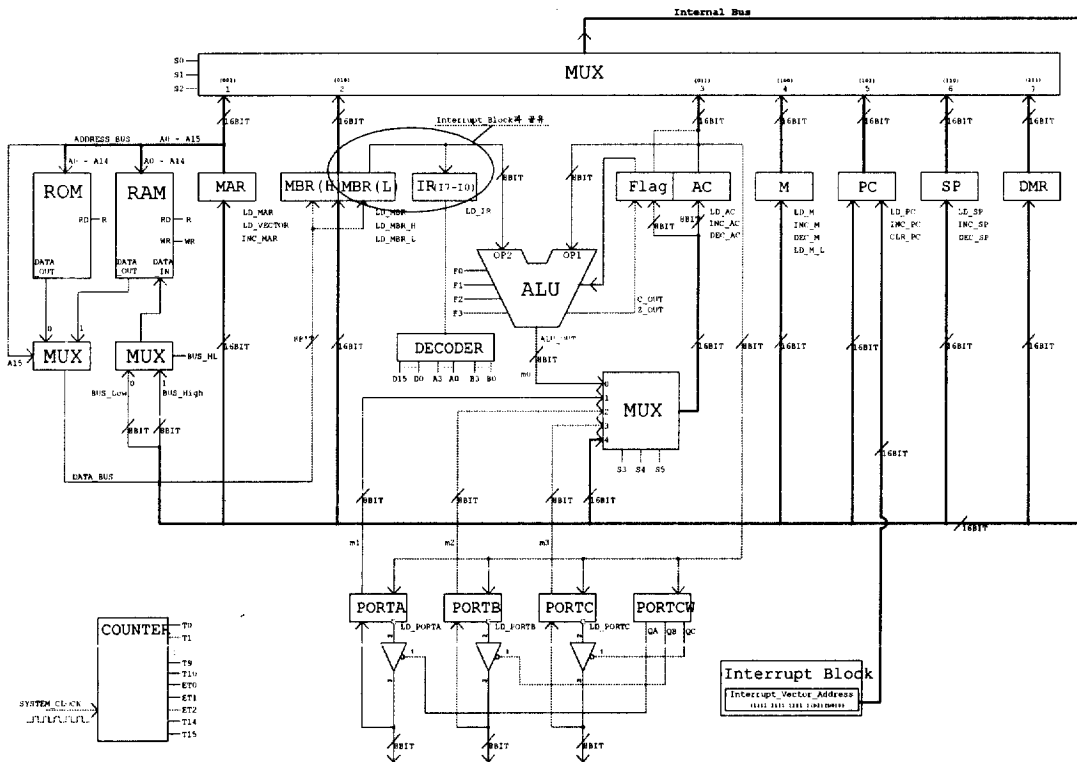
그림 1. 16비트 CPU Core블록도

그림 2. 인터럽트 하드웨어 구조
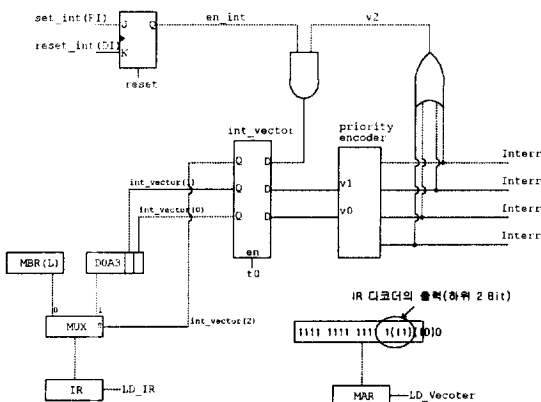
그림 3. 교배부(crossover part)

교부(compare part)를 사용하여 돌연변이율의 설정을 전용 CPU에서 가능하게 하였고 Decoder의 조정으로 돌연변이 될 비트의 선택도 자유롭게 하였다. 돌연변이 속도 역시 교배부와 마찬가지로 염색체를 16비트 단위로 1 Clock에서 돌연변이가 이루어지는 구조로 설계되었다.
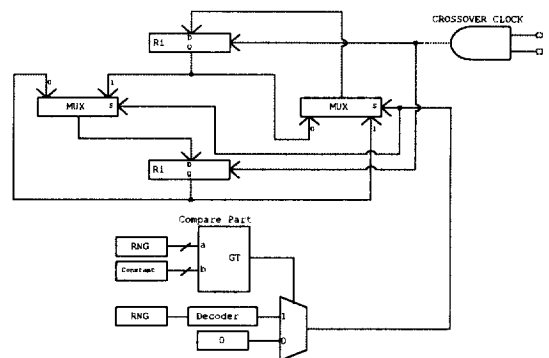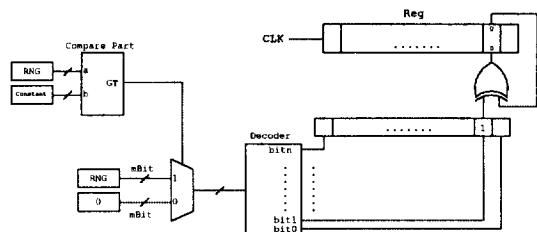
그림 4. 돌연변이부(Mutation part)

3.4 선택부(selection part)와 적합도 평가부 (fitness evaluation part)

본 논문에서는 하드웨어 구현에 적합한 정상상태 GA(steady-state GA)를 토너먼트 선택(tournament selection)과 함께 사용한다. 정상상태 GA(Steady-State GA)는 열성의 부모를 우성의 자손으로 대치하는 것이다. 이것은 세대모델(Generational GA)에 비해 메모리 사용량을 줄일 수 있다. 그리고 토너먼트 선택은 GA에서 일반적인 방법으로 사용되는 룰렛 선택(Roulette wheel selection)에 비해 하드웨어 자원을 줄인다는 면과 처리 속도에서 우수하다.

그림 5은 제안한 선택부(selection part) 와 적합도 평가부(fitness evaluation part)의 블록도이다. DMA로부터 전송한 4개의 염색체를 토너먼트로 선택하여 2개의 우성 염색체를 교배와 돌연변이를 하게된다. 16비트 단위를 단위 처리로 염색체를 처리하게 되는데 교배와 돌연변이 그리고 적합도 평가부에 이르기까지 Pipeline 처리를 하게 되어 있다. 16비트의 단위 처리는 32비트 이상의 처리 보다 속도 면에서 크게 뒤지지 않는다. 왜냐하면 HGA에서 속도저하를 가져오는 요인은 교배와 돌연변이 로직에 있지 않고 적합도를 평가하는 평가부에서 있기 때문이다. 제안한 적합도 평가부는 2개의 염색체를 동시에 처리 할 수 있게 이중 구조를 지니며 Pipeline 처리하여 DMA에 의해 외부 메모리에 저장시킨다.

4. 실험 및 고찰

기존에 연구된 유전자 알고리즘의 하드웨어 방식과 본 논문에서 제안한 방식을 비교하면 표 1과 같다. 표 1에서 보는 바와 같이 Scott의 연구 이후 지금까지의 유전자 알고리즘의 하드웨어 구현에 관한 연구에서 염색체의 길이는 고정되어 있음을 알 수 있다. 그리고 이 염색체의 고정으로 인하여 유전자 알고리즘의 실제적인 적용이 불가능하였다. 그러나 본 논문에서는 염색체를 16비트 단위로 Pipeline 처리함으로서 보다 실질적인 적용이 가능할 수 있다. 그리고 기존의 HGA는 HGA를 제어 할 수 있는 전용 CPU를 내장하고 있지 않음으로써 HGA의 제어를 위한 컴퓨터에의 의존은 불가피하였다. 그러나 본 논문에서는 HGA의 제어를 위한 16비트 CPU Core를 내장하고
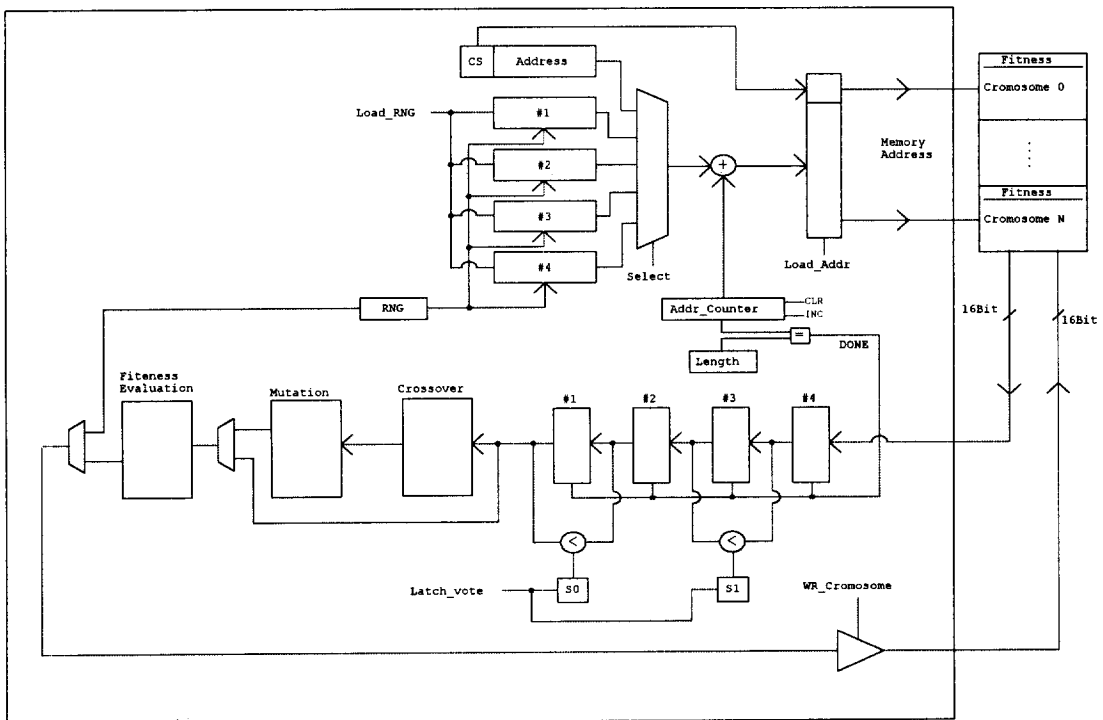
그림 5. DMA를 사용한 선택부와 적합도 평가부

표 1. 유전자 알고리즘의 하드웨어 구현에 관한 연구

	S.D. Scott[2] (HGA) Washington Univ 1994	N. Yoshida[3] (GAP) Kyushu Univ. 1997	T. Higuchi, I. Kajitani[13] Tsukuba 1998
Generation Steady-state	Steady-state	Steady-state	Steady-state
Selection	Roulette	Simplified tournament	Elitist Recombination
RNG	CA	CA	CA
Crossover point	1-point	1-point	Uniform
Chromosome. L	고정단위 3bit	고정단위 32bit	고정단위 2048bit
메인 컴퓨터에 독립된 전용 CPU의 내장 여부	없음	없음	없음
Note	BORG board	SFL(HDL)	EHW chip
	S. Wakabayashi[4] (GAA) Hiroshima Univ. 1998	D. J. Chung[5] (GAP) Inha Univ. 1999	제안된 원칩 컴퓨터방식 (HGA)
Generation Steady-state	Generation	Steady-state	Steady-state
Selection	Roulette & elitist strategy	Tournament	Tournament
RNG	CA	CA	CA
Crossover Point	2-point, Uniform, adaptive	1-point, 2-point, Uniform	1-point, 2-point, Uniform
Chromosome. L	고정단위 64bit	고정단위 Max 64bit	가변단위 16 x n bit
메인 컴퓨터에 독립된 전용 CPU의 내장 여부	없음	없음	있음
Note	0.5㎛ CMOS technology	Altera FPGA	Altera FPGA

있으므로 메인 컴퓨터에 독립적이다.

제안된 전용 원칩 컴퓨터의 효용성을 알아보기 위해서 하드웨어 진화에 적용하여 보았다. 그림 6은 제안된 전용 원칩 컴퓨터를 사용하여 비트 동기 회로를 진화시키기 위한 범용 State Machine이다. 이 범용 State Machine는 비트 동기회로인 동시에 평가회로부이다. 염색체는 ROM의 데이터가 되므로 유전자 연산을 위해 진화할 때는 RAM으로 설계하였다. 그림 6의 범용 State Machine에서 ROM의 내용에는 그림 7과 같은 유전자 알고리즘을 적용시킨 염색체가 저장된다. ROM의 주소입력은 입력 X와 함께 ROM의 출력 Q0-Q7이 피드백되어 플립플롭을 거쳐 입력된다. 그리고 ROM의 주소 입력에 의해 출력 Q0-Q8이 출력된다.

입력 X는 NRZ 랜덤 비트 신호를 입력시키는 단자이며 출력 Y는 입력 NRZ 랜덤 신호에 동기되어 입력비트 구간의 중간을 샘플링하게 하며 제안된 시스템으로 학습시킨다. 유전자 알고리즘을 적용시킨 집단수는 50개이며, 염색체 길이는 288비트이고 돌연변이율은 2%, 교배율은 80%이다.

그림 7은 비트 동기 추적 시스템을 구현하기 위해 유전자 알고리즘을 적용하여 출력한 적합도 99.88%의 염색체이다. 출력된 염색체는 비트 동기 추적 시스템을 위해 사용되며 ROM에 들어갈 데이터가 된다. 이 데이터는 ROM의 주소와 데이터 및 출력신호를 포함하고 있다.
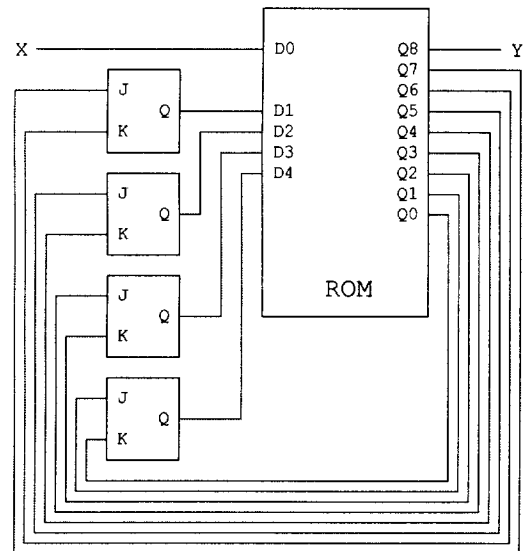
그림 6. 적합도 평가 회로

그림 8과 표 2는 제안된 시스템을 사용하여 비트 동기 추적 하드웨어의 세대수와 적합도의 관계를 나타낸 것이다. 최대 적합도 2500을 기준으로 세대수

```
0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1,
0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1,
1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0,
1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0,
1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1,
0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0,
1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0,
1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0,
1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1,
1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1,
1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1,
```

그림 7. 유전자 알고리즘에 의해 출력된 염색체
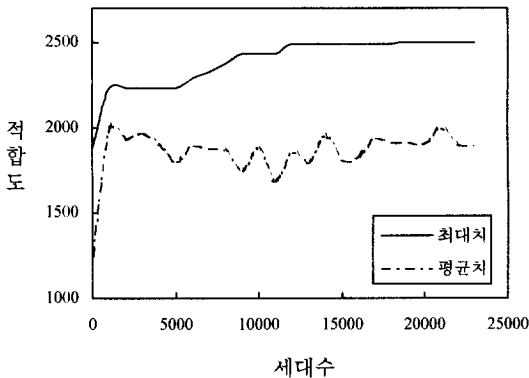
그림 8. 적합도와 세대수와의 관계

표 2. 세대수와 적합도와의 관계

세대수	최대 적합도	평균 적합도	세대수	최대 적합도	평균 적합도
1	1883	1254	12000	2493	1853
1000	2227	1993	13000	2493	1795
2000	2229	1933	14000	2494	1953
3000	2229	1970	15000	2494	1814
4000	2229	1897	16000	2494	1827
5000	2229	1805	17000	2494	1936
6000	2288	1894	18000	2494	1911
7000	2326	1882	19000	2497	1910
8000	2370	1869	20000	2497	1905
9000	2436	1744	21000	2497	1993
10000	2436	1878	22000	2497	1903
11000	2436	1690	25000	2497	1893

25000에 따른 적합도와 평균치의 변화를 표 2에서 알 수 있다. 유전자 알고리즘을 적용시켜 첫 세대에서 나온 적합도는 1883(75.32%)이며 세대수 1000에서 적합도가 급속하게 올라가는 것을 볼수 있으며 그때의 적합도는 2227(89.08%)이다. 그리고 세대수 19000부터 25000까지는 적합도가 2497(99.88%)이 계속해서 나온다. 제안된 시스템에서 비트 동기 추적 하드웨어를 진화 시켜본 결과 최대 적합도가 99.88%임을 알 수 있었다.

그림 9은 유전자 알고리즘을 적용한 비트 동기 추적 진화 하드웨어에서 입력 X에 NRZ 랜덤 신호를 입력하고 출력 Y로 비트 동기 추적 신호를 샘플링한 모습을 나타낸 것이다. 동기된 출력 신호는 NRZ의 입력 비트 구간이 시스템 클럭의 5 배일 때 입력 비트 구간의 중앙을 정확히 샘플링하도록 진화되었다. 실험 결과 지터가 없는 NRZ 랜덤 입력 신호 X에 대해 매우 잘 동작하는 것을 알 수 있었다. 그러나 입력 X의 비트 구간의 변화가 있을 때에는 불안정한 특성을 보이고 있음으로 보다 많은 연구가 필요하다.
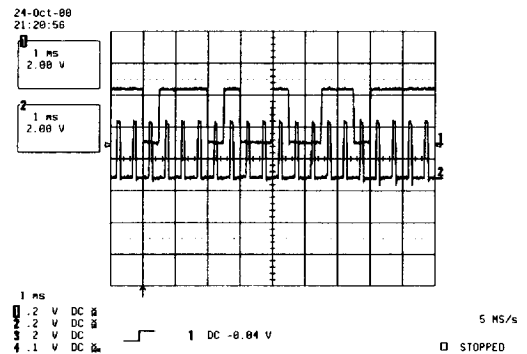
그림 9. NRZ 신호에 의한 비트 동기 파형

5. 결 론

유전자 알고리즘을 하드웨어로 처리하기 위해서 전용 원칩 컴퓨터를 제안하였다. 그리고 제안된 전용 원칩 컴퓨터를 Altera사의 FLEX10K100ARC로 구현하였다. 유전자 알고리즘 처리를 위한 전용 원칩 컴퓨터는 유전자 연산을 위한 하드웨어 HGA와 전용 16비트 CPU로 구성되어 있다. 전용 CPU는 16비트의 내부 버스와 인터럽트, DMA 있으며 유전자 연산을 위한 하드웨어인 HGA을 제어한다.

제안된 전용 원칩 컴퓨터는 염색체을 16비트 단위로 Pipeline 처리하며 교배, 돌연변이 그리고 적합도 평가부에 이르기까지 전과정이 Pipeline 처리를 하게 되어 있다. HGA에서 속도저하를 가져오는 요인은 교배와 돌연변이 로직에 있지 않고 적합도 평가부에서 있기 때문에 16비트의 단위처리는 32비트 이상의 처리 보다 속도 면에서 크게 뒤지지 않는다. 따라서 제안된 시스템은 염색체 길이에 무관하게 대규모 크기의 염색체를 처리할 수 있음으로 HGA의 실제 적용에 효과적이다. 그리고 설계된 전용 원칩 컴퓨터는 전용 CPU를 내장하고 있기 때문에 기존의 HGA와 달리 메인 컴퓨터에 독립적으로 동작된다.

제안된 시스템의 타당성을 알아보기 위해 하드웨어 진화(EHW: Evolvable Hardware)에 적용시켜 보았다. 그 결과 제안된 시스템에 의해 비트 동기 추적 하드웨어가 효과적으로 진화됨을 알 수 있었고 최종 진화된 비트 동기 추적 하드웨어의 최대 적합도가 99.88%임을 알 수 있었다. 진화된 비트 동기 추적 하드웨어는 지터가 없는 NRZ 램덤 비트 입력에 매우 좋은 특성을 보이나 지터가 있는 입력에는 불안정성이 보이므로 이를 보안할 연구가 필요하다.

참 고 문 헌

[1] J. Holland, "Adaptation in National and Artificial System", The University of Michigan, 1975., and MIT Press, 1992.

[2] S.D. Scott, A. Samal and S. Seth, "HGA:A hardware-based genetic algorithm", Proc. ACM/SIMDA 3rd International Symp. on FPGA, pp. 53-59, 1995

[3] N. Yosida, T. Moriki and T. Yasuoka, "GAP: Genetic VLSI processor for genetic algorithm", 1Second International ICSC Symp. on Soft Computing, pp. 341-345, 1997

[4] Shin'ichi Wakabayashi et al., "GAA:A VLSI genetic algorithm accelerator with on-the-fly adaptation of crossover operators", ISCAS 98, 1998

[5] Jin Jung Kim, Duck Jin Chung, "Implementation of Genetic Algorithm based on Hardware Optimization", TENCON '99 1999

[6] K. Dejong, An analysis of the behavior of class of genetic adaptive system, Ph.D Thesis, University of Michigan, 1975.

[7] Hiroaki Kitano, IDEN TEKI ALGOLITHM, SANGYO TOSHO, 1993.

[8] E. Vonk, L. C. Jain, and R. P. Johnson, Automatic Generation of Neural Network Architecture Using Evolutionary Computation, World Scientific Publishing, 1997.

[9] L. C. Jain, R. K. Jain, HYBRID INTELLIGENT ENGINEERING SYSTEMS, World Scientific Publishing, 1997.

[10] G. Sysweda, "Uniform Crossover in Genetic Algorithm", Proc. of ICGA-89, 1989.

[11] Melanie Mitchell, An introduction to Genetic Algorithm, The MIT Press, 1997.

[12] G. Winter et al., Genetic Algorithm in Engineering and Computer Science, John Wiley & Sons, 1996.

[13] I. Kajitani, T. Higuchi, "A gate-level EHW chip: Implementing GA operations and reconfigurable hardware on a signal LSI", Evolvable System: From Biology to Hardware, Lecture Notes in Computer Science 1478, pp. 1-12., Springer Verlag, 1998.

박 세 현

1980년 경북 대학교 공과대학 전자공학과 학사
1982년 경북 대학교 대학원 전자공학과 석사
1985년 아주 대학교 대학원 전자공학과 박사
1992년~현재 국립안동대학교 전자정보산업학부
1997년 11월 국민 포장 수여
1997년~1999년 국립 안동대학교 공과대학 학장
1999년~2000년 미시건 주립대학 전기 컴퓨터 공학과의 겸임 교수
2000년~현재 국립안동대학교 전자정보산업학부 부교수
관심분야 : 디지털시스템 설계, 컴퓨터구조, 유전자알고리즘

이 언 학

1997년 국립 안동대학교 공과대학 전자공학과 졸업.
2000년 국립 안동대학교 대학원 전자공학과 졸업
관심분야 : 컴퓨터구조, 유전자알고리즘